IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appellant: Stephen L. Bade et al.

Assignee: Synopsys, Inc.

Title: Method and System for Virtual Prototyping

Serial No.: 09/872,435 File Date: June 1, 2001

Examiner: Cuong V. Luu Art Unit: 2128

Docket No.: SYN-0472

August 7, 2007

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF

This Reply Brief is in response to the Examiner's Answer dated July 13, 2007.

/

/

/

/

/

/

/

/

/

/

/

Comments On Examiner's Answer

After a review of the arguments presented in the Examiner's Answer, it appears that the Examiner's Answer does not contain substantial new grounds of rejection.

However, to further highlight Appellant's position, Appellant comments herein on Figure 11 of Rompaey, which is used to support the Examiner's rejections of the claims. Specifically, Figure 11 is cited as teaching graphical symbols adapted to form a finite state machine (FSM) representation of electronic hardware. Appellant submits that Figure 11 is fundamentally hardware-centric and therefore does not teach an FSM as understood by those skilled in the art.

That is, Figure 11 shows various blocks (e.g. tracking & acquisition, frame extraction, correlator & noise estimator, chip matched filter, or A/D converter) and connections between such blocks. These blocks correspond to processes from Figure 10. However, Appellant submits that the presence of processes does not transform a hardware-centric implementation into a finite state machine implementation. Specifically, Figure 11 does not teach any states, which are integral to any basic FSM.

Appellant teaches in the Specification, paragraph 0014 (page 8, lines 10-17):

> One drawback of conventional electronic design approaches for embedded systems is that they are hardware-centric and typically have an execution speed that is too slow for the simulation to serve as a hardware emulator for the purposes of debugging software. For example, one conventional electronic design method is to use a hardware definition language (HDL) to model the hardware partition at the hardware implementation level. However, modeling the hardware implementation in HDL requires that the hardware be extensively developed (e.g., fully detailed) before

the software, which can require a significant length of time in the development cycle.

Fig. 17A shows a conventional finite state machine representation of a hardware peripheral in a transition condition/action format. Paragraph 0040 (page 16, lines 1-2).

In contrast, in accordance with the present invention, Appellant teaches in the Specification, paragraph 0128 (page 52, line 12 to page 53, line 7):

> The behavior of a Process is described as a FSM. When started, a Process executes its start transition and enters the first state. The reception of a signal triggers the transition from one state to the next state. In transitions, a Process may execute Actions. Actions can assign values to variables, perform calculations, branch on values of expressions, call functions and procedures, and send signals to other Processes. A Process can define local variables, defined by the scope of the Process, by means of a Declaration. Initialization places all the Process FSMs into the Start Construct, as depicted by an oval. At that point, the Process executes the action of its start transition, as described in the Task Construct (depicted by the rectangle). In a preferred embodiment, a designer specifies the Task Construct behavior using plain (ANSI) C++, minimizing the learning curve typically associated with a new language or paradigm. Each Process receives (and keeps) signals in an (implicit) queue. In a State, depicted by a rectangle with rounded sides, the Process takes the first signal from the queue. In the example, the interrupt controller waits for an interrupt signal, as sent by one of the two peripheral devices. The Symbol for a Signal-In construct is a rectangle with an arrow pointing inward as either its left or right side.

For example, Fig. 17B shows the FSM of Fig. 17A written using graphical constructs in accordance with the present invention. Paragraph 0041 (page 16, lines 3-4)(note state

symbols 1908, signal_out symbols 1912, and signal_in symbols 1914)(Figs. 19A and 19B illustrating exemplary symbols). Fig. 18 illustrates two communicating FSMs with implicit signal queues explicitly drawn. Paragraph 0042 (page 16, lines 5-6)(note signal_in symbols, signal_out symbols, and state symbols s_0, s_1, s_2). Fig. 20 is a screenshot showing an example of an FSM representation of a hardware counter peripheral. Paragraph 0044 (page 16, lines 9-10)(note various non-process/task symbols, e.g. signal_out, signal_in, decision, start, and state symbols).

Advantageously, unlike hardware-centric embodiments, FSMs representations facilitate an extremely fast simulation. Paragraph 0095 (page 29, lines 1-9).

Based on the above comments, Appellants submit that Figure 11 of Rompaey fails to disclose or suggest an FSM representation of electronic hardware.

The Examiner also characterizes col. 9, lines 14-27 of Rompaey as teaching a processor made up of a control unit and an arithmetic logic unit, wherein the control unit is divided into 2 sub-units (one of which is a FSM). Appellant cannot find support for these characterizations in the cited passage. Appellant notes that the Examiner previously cited col. 9, lines 19-22 in his rejections. However, the additional passages do not add substantively to the previously presented arguments. Therefore, Appellant's response to the expanded citation is substantially the same for the original citation.

For the additional foregoing reasons, it is submitted that the Examiner's rejections of Claims 37-72, 74-78, and 89 are erroneous, and reversal of these rejections is respectfully requested.

Respectfully submitted,

Customer No.: 35273

Jeanette S. Harms
Attorney for Appellant
Reg. No. 35,537

Telephone: 408-451-5907
Facsimile: 408-451-5908